



**University of  
Zurich**<sup>UZH</sup>

**Zurich Open Repository and  
Archive**

University of Zurich  
University Library  
Strickhofstrasse 39  
CH-8057 Zurich  
[www.zora.uzh.ch](http://www.zora.uzh.ch)

---

Year: 2014

---

## **Quest for Requirements: Scrutinizing Advanced Search Queries for Cloud Services with Fuzzy Galois Lattices**

Todoran, Irina ; Glinz, Martin

**Abstract:** In software and requirements engineering, requirements elicitation is considered an essential step towards building successful systems. Despite extensive existing research in the field of distributed requirements engineering, the topic of requirements elicitation for cloud systems remains still uncovered. Cloud challenges (e.g., heterogeneous and globally distributed users, volatile requirements, frequent change requests) cannot always be satisfied by existing methods. We present a new approach for eliciting requirements for cloud services by analyzing advanced search queries. Our approach builds fuzzy Galois lattices for the terms that compose advanced search queries, thus enabling a thorough analysis of stored search data. This can support cloud providers in observing requirements clusters and new classes of cloud services, identifying the threshold for achieving satisfied consumers with a minimal set of requirements implemented, and thus designing novel solutions, based on market trends. Moreover, the Galois lattices approach enables large-scale consumers' involvement and ensures the elicitation of real requirements unobtrusively.

DOI: <https://doi.org/10.1109/SERVICES.2014.49>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-99276>

Conference or Workshop Item

Published Version

Originally published at:

Todoran, Irina; Glinz, Martin (2014). Quest for Requirements: Scrutinizing Advanced Search Queries for Cloud Services with Fuzzy Galois Lattices. In: IEEE 10th World Congress on Services 2014 (SERVICES 2014), Anchorage, AK, USA, 27 June 2014 - 2 July 2014. IEEE, 234-241.

DOI: <https://doi.org/10.1109/SERVICES.2014.49>

# Quest for Requirements: Scrutinizing Advanced Search Queries for Cloud Services with Fuzzy Galois Lattices

Irina Todoran, Martin Glinz

Department of Informatics, University of Zurich, Switzerland  
{todoran, glinz}@ifi.uzh.ch

**Abstract**—In software and requirements engineering, requirements elicitation is considered an essential step towards building successful systems. Despite extensive existing research in the field of distributed requirements engineering, the topic of requirements elicitation for *cloud systems* remains still uncovered. Cloud challenges (e.g., heterogeneous and globally distributed users, volatile requirements, frequent change requests) cannot always be satisfied by existing methods. We present a new approach for eliciting requirements for cloud services by analyzing advanced search queries. Our approach builds fuzzy Galois lattices for the terms that compose advanced search queries, thus enabling a thorough analysis of stored search data. This can support cloud providers in observing requirements clusters and new classes of cloud services, identifying the threshold for achieving satisfied consumers with a minimal set of requirements implemented, and thus designing novel solutions, based on market trends. Moreover, the Galois lattices approach enables large-scale consumers' involvement and ensures the elicitation of real requirements unobtrusively.

**Keywords**—cloud computing; requirements elicitation; advanced search query; Galois lattice; data analysis;

## I. INTRODUCTION

Requirements elicitation, that is seeking, capturing and consolidating requirements, is a core activity in any requirements engineering (RE) process [1]. Therefore, numerous elicitation techniques have been developed and are in use nowadays [2]. Using elicitation techniques that do not fit the characteristics of the project increases RE costs and makes the project failure-prone [3], [4]. Hence, approaches have been investigated for assigning techniques to contexts and selecting appropriate methods for individual cases [3]. Researchers have also provided comparisons [2] and best practices on how to use these methods [1].

However, most existing techniques mainly address settings where stakeholders can be identified and analysts can directly interact with them. In today's context of *cloud systems* [5], traditional requirements elicitation techniques are heavily challenged [6]. For instance, the communication with consumers becomes too expensive or even impossible because the key consumers are no longer known in person, being both too numerous and too heterogeneous. Since cloud consumers are often also globally distributed, it is virtually impossible to consider specific individual stakeholders.

Despite the rapid growth of the cloud use and the high number of cloud services available, dedicated requirements

elicitation methods for the cloud are lacking [7]. As a result, cloud service providers have tried to adapt traditional methods such as workshops and interviews to work in distributed settings, e.g., by organizing online workshops and VoIP interviews supported by rich media [8]. Others have used artificial stakeholders invented by marketing or substituted the global stakeholder community by a few pilot customers. Nevertheless, our previous research shows that such approaches have been rather unsuccessful so far [8]. This finding is also supported by other researchers, who consider that the existing methods provide insufficient support or are difficult to apply in practice [9], [10]. Therefore, the cloud calls for thoroughly different requirements elicitation methods.

To address this research gap, we are investigating the possibility to infer new cloud service requirements from advanced search queries performed by (potential) cloud consumers. An *advanced search query* is a query that goes beyond simple keyword search by providing search information in some structured form. We exploit a particular form of advanced search queries where, for a given set of service features, users specify desired values for all those features (see Sect. III B for an example). The data can be collected either on cloud providers' websites or on platforms that aggregate services from multiple cloud suppliers (marketplaces [11]), provided that they expose advanced search capabilities for the services available.

In this paper, we present our approach. We consider a given set  $S$  of search queries, each of them specifying desired values for a set  $F$  of service features. We first convert this information into a fuzzy binary relation  $\tilde{R}(S, F)$ . From this relation, we construct a fuzzy Galois lattice [12], which can then be analyzed with respect to three purposes: (i) understanding how requirements of (potential) service consumers can be clustered, (ii) identifying the threshold for achieving satisfied consumers with a minimal set of requirements implemented, and (iii) identifying new classes of cloud services, based on market trends. Our approach contributes a novel requirements elicitation technique which is specifically tailored to a cloud computing context.

The remainder of this paper is organized as follows. Section II summarizes the related work in the area of requirements elicitation techniques, potentially applicable

in the cloud. Section III presents our new approach and applies it on a concrete example. The outcome is discussed in Section IV, and Section V concludes the paper.

## II. RELATED WORK

From the early 2000s, researchers observed that requirements engineering also needs to consider distributed [13] and asynchronous settings [14], and this currently extends to the cloud context. However, due to its collaboration-intensive and time-consuming nature, requirements elicitation becomes difficult in the cloud [13], [15].

As far as *dedicated cloud requirements elicitation methods* are concerned, there has been some advancement during the recent years. For instance, frameworks focusing on the supply-demand relation have been designed [16], and management systems for requirements ensuring QoS have been developed [17]. Moreover, researchers looked into methods for eliciting particular types of requirements, e.g., legal [18] or security [19]. Still, these are only niche recommendations and no comprehensive clear solution exists, addressing the cloud-specific requirements elicitation challenges.

As far as *distributed requirements elicitation* is concerned, Lloyd et al. conducted a study [10] on the effectiveness of elicitation techniques in distributed requirements engineering, concluding that synchronous elicitation approaches are generally more effective than asynchronous ones. Lim et al. [20] present ideas on asynchronous and distributed stakeholder identification, assuming that key stakeholders are known, and further users can be identified based on domain knowledge. However, such approaches do not easily extend to the cloud context, since the audience for services is most often unknown and globally distributed.

Tuunanen [4] addresses the problem of reaching and involving wide audience end-users, or users who are not within organizational reach. He argues that traditional techniques do not provide adequate solutions and presents methods which could potentially fill this gap (e.g., EasyWinWin). However, none of these methods has been successfully used on a large scale for distributed elicitation so far. Moreover, research on EasyWinWin by Kukreja et al. [21] promises to provide support for distributed settings, but only focuses on stakeholders within organizational reach.

Another challenge of requirements elicitation in the cloud is the continuous change of consumer needs [22]. Consequently, various wiki approaches have been implemented, to provide a time-efficient possibility for updating and eliciting requirements. For instance, Decker et al. developed a wiki-based solution that enables stakeholders' participation in RE [23], Solis and Ali's spatial hypertext wiki focuses on distributed teams [24], whereas Liang et al. [25] and Lohmann et al. [26] exploit semantic annotation wikis. However, wiki-based methods generally assume that stakeholders are at least identifiable, which is not the case in a cloud context.

Studies from the field of web-information systems by Yang and Tang [9] reveal that the elicitation needs regarding Internet-based systems are also rather different from those of traditional systems (e.g., due to higher user diversity). Moreover, most existing requirements elicitation methods can only deal with a limited number of stakeholders [22]. The number of potential cloud service consumers may often go beyond what traditional methods can handle [19], and no real solutions addressing this elicitation issue have been developed so far. Market-driven techniques [27], which are usually employed when it is impossible to consider individual consumers, prove to be rather limited in the cloud, due to the lack of specific localized markets.

Work in data mining, machine learning and particularly recommender systems [28] also addresses the problem of extracting value from search data. For example, search-based and collaborative techniques can make personalized online product recommendations [29], and user feedback has been used to rank various products [30]. Throughout the recent years, recommender systems [31] (e.g., probability-based collaborative filtering [32]) and clustering data mining methods [33] have been heavily used for marketing purposes, to suggest similar products in e-commerce systems, or to segment populations. However, to our knowledge, such techniques have never been adapted or utilized for requirements elicitation in the cloud.

To summarize, the existing elicitation techniques, even when adapted, are mostly unsuitable for the cloud, and can support cloud service providers only to a limited extent in their requirements elicitation processes.

## III. OUR NEW APPROACH: FUZZY GALOIS LATTICE ANALYSIS FOR CLOUD REQUIREMENTS ELICITATION

Based on the existing related work and our previous research [8], we found there is an evident need for dedicated requirements elicitation methods for the cloud. These should meet the following requirements:

- R1: Fit for wider and heterogeneous audiences;
- R2: Take less time than traditional elicitation methods;
- R3: Make automated elicitation possible;
- R4: Be applied remotely;
- R5: Be able to handle volatile requirements.

To satisfy these requirements, we propose analyzing the data collected by cloud service providers or marketplaces in the form of advanced search queries, to infer new consumer requirements. For this, we represent the search queries by a fuzzy Galois lattice [12]. Galois lattices allow the identification of sensible groupings of objects with common attributes. Due to these clustering and hierarchy properties [34], Galois lattices have been used in software engineering for object identification [35], software modularization and analysis [36], and browsing software component libraries [37]. However, to our knowledge, these properties have not

been explored for new requirements acquisition based on advanced search queries.

### A. Mathematical Foundations

In this sub-section, we briefly review the mathematical foundations needed for our approach.

A partially ordered set (poset)  $A$  is called a *lattice* iff for any subset  $A'$  of  $A$ , there exist a least upper bound  $\sup \in A$  (the *supremum* of  $A'$ ) and a least lower bound  $\inf \in A$  (the *infimum* of  $A'$ ). The supremum of  $A'$  is the smallest element of  $A$  that is greater than or equal to each element of  $A'$ . It is unique and it may or may not belong to  $A'$ . The infimum of  $A'$  is defined analogously. Lattices can be graphically represented as acyclic directed graphs having exactly one source node (with no incoming edges) and one sink node (with no outgoing edges).

Galois connections have their roots in Galois theory [38], [39], and refer to correspondences between two partially ordered sets (posets). If  $(A, \leq)$  and  $(B, \leq)$  are two posets, a monotone Galois connection between them consists of two monotone functions  $F : A \rightarrow B$  and  $G : B \rightarrow A$ , such that for all  $a \in A$  and  $b \in B$ , we have  $F(a) \leq b$  iff  $a \leq G(b)$ . The posets can be represented hierarchically in a graded system of sub- and superconcepts, which follows the mathematical axioms of a lattice. In a Galois concept lattice, the elements can generally take binary values. For a detailed overview on Galois theory, refer to [38], [39].

A *binary relation*  $R(X, Y)$  is a set of ordered pairs  $(x, y)$ ,  $x \in X, y \in Y$ . For any given elements  $p \in X$  and  $q \in Y$ , the pair  $(p, q)$  is either an element of  $R(X, Y)$  or it is not. *Fuzzy binary relations*  $\tilde{R}$  extend this digital behavior by allowing a *degree of membership* in a relation: the degree of membership of  $(p, q)$  in  $\tilde{R}$  may be any real number from the interval  $[0, 1]$ . Obviously, the special case where every pair has a membership value of either zero or one represents a normal (crisp) binary relation.

In contrast to general formal concept analysis theory (FCA) [40], Galois connections take into consideration the relations between fuzzy concepts represented on ratio scales. For example, 0.2 and 0.5 are only two distinct values in FCA, whereas 0.2 and 0.5 are two values which can be ordered in Galois connections theory, e.g.,  $0.2 < 0.5$ . This leads to the notion of fuzzy Galois lattices [12] the nodes of which represent fuzzy concepts, which in turn are constructed from a fuzzy binary relation.

### B. A Running Example

We illustrate our approach with a concrete example of advanced searches for cloud data storage services, based on ten features. The first two columns of Table II list the features considered. While performing advanced search queries, consumers specify values for these features, based on their needs.

Table I  
FUZZY GALOIS LATTICE ANALYSIS FOR CLOUD REQUIREMENTS  
ELICITATION - SUMMARY

Step	Description of the step
1.	Collect advanced search queries for cloud services via a search platform.
2.	Model the search queries as formal concepts.
3.	Represent query data as a fuzzy binary relation.
4.	Calculate all the FCs for the given set of service queries.
5.	Analyze the fuzzy binary relation for special properties, apply reductions where possible.
6.	Represent the FCs in a fuzzy Galois concept lattice.
7.	Analyze the lattice nodes, the supremum and infimum elements for sub-lattices and potential clusters leading to new requirements for cloud services.

When we model cloud service queries, some features can be easily represented using only binary values, e.g., the service provides mobile support (1) or not (0). However, numerous features are better represented on ratio scales, e.g., for data storage cloud services, the values can be between 1 GB and 20 TB; in this case, binary values would be difficult to use. Therefore, we use the extension of the Galois lattice theory to fuzzy binary relations [12], such that features can not only be represented on a nominal scale (taking the binary values 0 or 1), but also on a ratio scale. Therefore, a fuzzy set  $S_i$  includes a degree of membership for each of its elements, taking a value in the range  $[0, 1]$ . A set with the membership degrees restricted to the values 0 and 1 (crisp set) is a particular type of a fuzzy set, so it is formally correct to mix the features on a nominal scale with those on a ratio scale in the same representation.

For example, the feature “ $f_1$ : Private user” is represented on a nominal scale (N) and can take the value 0, if the service is not available for private users, or 1 if the service is offered for private users. The feature “ $f_3$ : Storage” is represented on a ratio scale (R) and can take fuzzy values in the range  $[0, 1]$ .

### C. The Steps of the Approach

In this sub-section, we present the mechanics of our approach, consisting of seven steps. For each step, we first explain how it works in theory, and then apply it to our running example.

Conceptually, each advanced search query is composed of a set of cloud service features, which are specified by a (potential) consumer who is searching for a cloud service. In Galois theory, such a set is called a formal concept (FC). All search queries of a considered dataset are modelled as FCs in a Galois lattice, similar to the nodes of a graph. Moreover, the supremum and infimum elements of sub-lattices of the main lattice are calculated and also represented as FCs.

A fuzzy Galois lattice of a set of advanced search queries for cloud services can be generated and analyzed in seven steps, as follows. These are also summarized in Table I.

**Step 1.** Having access to a search platform with advanced search capabilities and pre-defined possible features and values, data is collected from (potential) cloud consumers. Service features may include both functional (e.g., storage) and non-functional requirements (e.g., reliability).

*Example:* we collect advanced search queries for data storage services, using a marketplace for cloud services [41]. This allows (potential) cloud service consumers to input their needs using predefined advanced search criteria.

**Step 2.** The search queries are modeled as fuzzy FCs using monotonic modeling functions.

*Example:* let  $S$  be a set (the universe of discourse) that denotes a generic data storage cloud service, with ten predefined features on the search platform, that users can opt for, as shown in Table II. For each of these features, cloud providers can define monotonic modeling functions  $f(x)$  that transform the numerical values input by users into fuzzy values. For example, for the feature “storage capacity” of a cloud storage service, we can have a monotonic function as follows:

$$f(x) = \begin{cases} 10^{-3}x, & x < 10^3 \text{ GB} \\ 1, & x \geq 10^3 \text{ GB} \end{cases}$$

Accordingly, a value of 500 GB will be transformed into the fuzzy value 0.5. For features represented on a nominal scale, such as “AES encryption”,  $f(x)$  can take the value of 1 if the feature is available, else 0. The rest of features work similarly. Naturally, there are numerous ways in which  $f(x)$  can be defined; the choice only has to ensure that it is a monotonic transformation which leads to values in the range [0,1], and maintains a ratio scale for likely fuzzy values. Then, an advanced query for a cloud service can be defined as a fuzzy set  $S_i = \{f_{ij} : j = 1, n\}$ , where  $f$  represents the features of the cloud service, and  $n$  is the number of features defined in the search platform for the generic type of service  $S$ .

**Step 3.** The data is represented in a matrix, as a fuzzy binary relation  $\tilde{R}(S, F)$ .

*Example:* our data consists of five queries, represented as  $\tilde{R}(S, F)$  in Table III. The search queries are shown as rows in the table:  $S_i$ ,  $i=1,5$ . For instance,  $S_2 = \{f_1/0, f_2/1, f_3/0.2, f_4/1, f_5/0.9, f_6/0.8, f_7/1, f_8/0, f_9/0.5, f_{10}/1\}$  is an example of a fuzzy set representing an advanced query for a service, with the fuzzy values  $\{0, 1, 0.2, 1, 0.9, 0.8, 1, 0, 0.5, 1\}$ . In practice, this means that a (potential) consumer made an advanced search for a data storage service which is available only for business users and not for private users, can store up to 200 GB, provides mobile support, can recover files which are up to 90 days old, has a reliability of 98%, uses AES encryption but does not use SSL encryption, the maximum size per file is 5 GB and has an uptime higher than 99%.

**Step 4.** For all elements of the power set  $\mathcal{P}(S)$  of the set  $S$  of search queries, we calculate the fuzzy concepts

Table II  
SERVICE FEATURES (N = NOMINAL SCALE: {0,1},  
R = RATIO SCALE: [0,1])

No.	Feature	Scale	Modeling function $f(x)$
$f_1$	Private user	N	0: N/A, 1: available
$f_2$	Business user	N	0: N/A, 1: available
$f_3$	Storage	R	$10^{-3}x, x < 10^3 \text{ GB};$ $1, x \geq 10^3 \text{ GB}$
$f_4$	Mobile support	N	0: N/A, 1: available
$f_5$	File recovery	R	$10^{-2}x, x < 10^2 \text{ days};$ $1, x \geq 10^2 \text{ days}$
$f_6$	Reliability	R	$0, x \leq 90\%; 1, x > 99\%;$ $0.1 * (x - 90), x \in (90, 99\%]$
$f_7$	AES encryption	N	0: N/A, 1: available
$f_8$	SSL encryption	N	0: N/A, 1: available
$f_9$	Max size/file	R	$0, x < 0.1 \text{ GB}; 1, x \geq 10 \text{ GB}$ $0.1 * x, x \in [0.1, 10) \text{ GB}$
$f_{10}$	Uptime	R	$0, x \leq 90\%; 1, x > 99\%;$ $0.1 * (x - 90), x \in (90, 99\%]$

Table III  
THE FUZZY BINARY RELATION  $\tilde{R}(S, F)$

$\tilde{R}$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$
$S_1$	1	1	0.7	0	0.3	0.4	1	0	0.3	1
$S_2$	0	1	0.2	1	0.9	0.8	1	0	0.5	1
$S_3$	0	1	0.1	0	0.5	0	1	0	0	0.9
$S_4$	1	0	1	0	0.9	0.3	1	1	0.8	0.7
$S_5$	1	0	0.5	0	0.3	0.7	0	0	0.6	1

FC, yielding  $2^n$  FCs, where  $n$  is the number of queries. According to Galois connections theory, the FC belonging to a subset  $S'$  is calculated by taking the minima of all feature values of the queries contained in  $S'$ .

*Example:* we compute the complete list<sup>1</sup> of  $2^5 = 32$  fuzzy formal concepts FC. Due to space constraints, only a partial list is shown in Table IV - for the complete list, please refer to the link. The indices indicate the queries that each FC is constructed of, e.g.,  $FC_{1,2,4}$  is a formal concept constructed of  $FC_1$ ,  $FC_2$  and  $FC_4$ , through intermediary formal concepts  $FC_{1,2}$ ,  $FC_{1,4}$  and  $FC_{2,4}$ .

**Step 5.** In case the context (matrix) exposes special properties, these are considered at this stage. As a typical example, assume we detect a small distance between two rows of the matrix. Since this method is based on computing minimum values, detecting a search query which is the minimum of another will lead to reduction opportunities in the final lattice, i.e. some nodes do not have to be represented due to redundancy. Another example is if we have duplicate entries in the list of computed FCs.

<sup>1</sup>[http://www.ifi.uzh.ch/rerg/people/todoran/FC\\_Complete\\_List.pdf](http://www.ifi.uzh.ch/rerg/people/todoran/FC_Complete_List.pdf)

Table IV  
FUZZY CONCEPTS CALCULATED FROM  $\tilde{R}$  (PARTIAL LIST)

Label	Fuzzy concept
$FC_\emptyset$	$\{f_1/1, f_2/1, f_3/1, f_4/1, f_5/0.9, f_6/0.8, f_7/1, f_8/1, f_9/0.8, f_{10}/1\}$
...	
$FC_4$	$\{f_1/1, f_2/0, f_3/1, f_4/0, f_5/0.9, f_6/0.3, f_7/1, f_8/1, f_9/0.8, f_{10}/0.7\}$
$FC_5$	$\{f_1/0, f_2/0, f_3/0.5, f_4/0, f_5/0.3, f_6/0.7, f_7/0, f_8/0, f_9/0.6, f_{10}/1\}$
...	
$FC_{4,5}$	$\{f_1/1, f_2/0, f_3/0.5, f_4/0, f_5/0.3, f_6/0.3, f_7/0, f_8/0, f_9/0.6, f_{10}/0.7\}$
...	
$FC_{1,4,5}$	$\{f_1/1, f_2/0, f_3/0.5, f_4/0, f_5/0.3, f_6/0.3, f_7/0, f_8/0, f_9/0.3, f_{10}/0.7\}$
$FC_{2,4,5}$	$\{f_1/0, f_2/0, f_3/0.2, f_4/0, f_5/0.3, f_6/0.3, f_7/0, f_8/0, f_9/0.5, f_{10}/0.7\}$
...	
$FC_{1,2,4,5}$	$\{f_1/0, f_2/0, f_3/0.2, f_4/0, f_5/0.3, f_6/0.3, f_7/0, f_8/0, f_9/0.3, f_{10}/0.7\}$
...	
$FC_{1,2,3,4,5}$	$\{f_1/0, f_2/0, f_3/0.1, f_4/0, f_5/0.3, f_6/0, f_7/0, f_8/0, f_9/0, f_{10}/0.7\}$

*Example:* analyzing the matrix  $\tilde{R}$ , we notice that the minimum of service queries  $S_2$  and  $S_3$  is  $S_3$ , at a small distance for some of the features. This leads to opportunities for reduction, since some FCs will have identical values. Therefore, we will generate a sub-lattice corresponding to  $\tilde{R}$ , where each distinct node appears only once, as shown in the FCs list resulted after eliminating duplicates<sup>2</sup>, e.g.,  $FC_3 = FC_{2,3}$ . Further, analyzing the complete list of FCs generated, several other duplicates can be identified. For example,  $FC_{1,3} = FC_{1,2,3}$ , since the minimum of  $S_2$  and  $S_3$  is  $S_3$ . Moreover,  $FC_{3,4} = FC_{2,3,4}$ ,  $FC_{3,5} = FC_{1,3,5} = FC_{2,3,5} = FC_{1,2,3,5}$ ,  $FC_{1,3,4} = FC_{1,2,3,4}$  and  $FC_{3,4,5} = FC_{1,3,4,5} = FC_{2,3,4,5} = FC_{1,2,3,4,5}$ .

**Step 6.** We represent the fuzzy FCs in a concept lattice.

*Example:* the unique fuzzy formal concepts resulted in Step 5 are graphically represented as a fuzzy Galois sub-lattice of  $\tilde{R}$ , as shown in Figure 1.

**Step 7.** Finally, we analyze the lattice: if/how the FCs cluster, new feature combinations potentially leading to new services that could be developed and that do not exist at the moment, the supremum and infimum for sub-lattices leading to new ideas for cloud services, that satisfy significant populations.

*Example:* the output of our approach is analyzed in the following section.

#### D. Analysis of the Results

If a cloud provider wanted to fully satisfy only one query, i.e. satisfy all the features requested, exactly to the extents requested, it would have to supply a service having exactly the features specified in the query. However, this is unreasonable in most situations, since cloud providers cannot take into account individual wishes from each (potential)

consumer, but rather target groups of consumers. Using our approach, for every subset of FCs, the supremum and infimum elements can be calculated. Practically speaking, the supremum of a set of FCs represents a comprehensive service that satisfies all the features requested in the corresponding subset of queries. For instance, if the cloud provider decides to consider all five queries, he could use the supremum of the queries, which is  $FC_\emptyset$ . Nevertheless, despite satisfying all queries, this may be impossible or too costly to implement. Therefore, we will analyze how the queries can be clustered and what minimum combinations of features would still achieve satisfied populations. For this, we analyze the infimum elements of sub-lattices. The infimum represents a cloud service which fully satisfies only those features that all queries in the corresponding subset have in common, while the rest are partially or not satisfied.

Since the space would not allow a complete analysis of the lattice generated in Step 6 (Figure 1), we choose to analyze a sub-lattice  $SL$ , e.g., for service queries  $S_4$  and  $S_5$ . The relevant formal concepts for this sub-lattice are: the empty set  $FC_\emptyset$ ,  $FC_4$  and  $FC_5$  which are a 1:1 mapping of the advanced service queries, and the infimum nodes computed by our approach, which include both queries  $S_4$  and  $S_5$ :  $FC_{4,5}$ ,  $FC_{1,4,5}$ ,  $FC_{2,4,5}$ ,  $FC_{1,2,4,5}$  and  $FC_{1,2,3,4,5}$ . These are colored in grey in Figure 1.

For queries  $S_4$  and  $S_5$ , we calculate how many features are fully and partially satisfied by the infimum elements of  $SL$ . The graph in Figure 2 shows that  $FC_{4,5}$  fully satisfies five out of ten features for  $S_4$ :  $FC_{4,5}$  models a service that is available for private users and not for business users, does not provide mobile support, has 93% reliability and 97% uptime.  $S_5$  is fully satisfied to a higher rate, with eight out of ten features:  $f_1 - f_5$  and  $f_7 - f_9$ . If we cumulate the fully and partially satisfied features, we find that  $FC_{4,5}$  satisfies eight features for  $S_4$  and all features for

<sup>2</sup>[http://www.ifi.uzh.ch/terg/people/todoran/Unique\\_FC\\_List.pdf](http://www.ifi.uzh.ch/terg/people/todoran/Unique_FC_List.pdf)

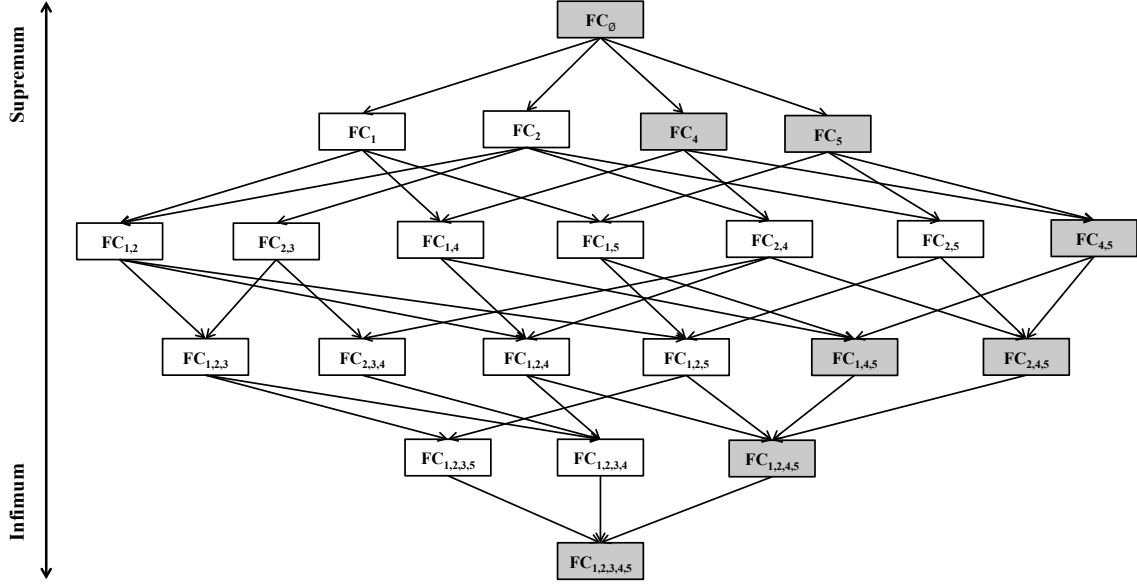


Figure 1. Fuzzy Galois sub-lattice of  $\tilde{R}$ . The grey FCs represent the sub-lattice  $SL$  discussed in Sec. III D.

$S_5$ , respectively. Depending on the provider's strategy, this may already be a good compromise to address queries  $S_4$  and  $S_5$ . Continuing the analysis, we observe that  $FC_{1,4,5}$  does not differ significantly, since it fully satisfies the same features as  $FC_{4,5}$  for  $S_4$ , and only changes the extent to which  $f_9$  is satisfied for  $S_5$ : instead of allowing a maximum file size of 6 GB,  $FC_{1,4,5}$  only allows files of maximum 3 GB. As far as the cumulated satisfaction is concerned, this is identical to the one reached with  $FC_{4,5}$ . Moreover, since  $FC_{1,4,5}$  is also an infimum for  $FC_1$ , it will satisfy some of the features mentioned in  $S_1$ : five fully and three partially. Therefore, given the overall performance, it seems  $FC_{1,4,5}$  would be a better choice than  $FC_{4,5}$ , if the cloud provider decided to supply a new service addressing service queries  $S_4$  and  $S_5$ . Similarly, the fitness rates for  $FC_{2,4,5}$ ,  $FC_{1,2,4,5}$  and  $FC_{1,2,3,4,5}$  are analyzed. The graph shows that  $FC_{2,4,5}$  is less suitable than  $FC_{1,4,5}$ , when aiming to satisfy the authors of queries  $S_4$  and  $S_5$ , with an advantage of only four fully and three partially satisfied features for  $S_2$ .  $FC_{1,2,4,5}$  is equal in performance to  $FC_{2,4,5}$  as far as  $S_4$  and  $S_5$  are concerned, but with a higher advantage: it satisfies four features fully and three partially for  $S_1$ , and also satisfies three features fully and four partially for  $S_2$ . Eventually, the number of features satisfied converges to zero, reaching zero if the input queries represent disjoint sets.

This analysis shows that  $S_4$  and  $S_5$  can be satisfied simultaneously by several combinations of features, which are more economical to implement than services addressing individual needs, still maintaining high satisfaction rates. Infimum elements representing new classes of services such as  $FC_{4,5}$ ,  $FC_{1,4,5}$  and even  $FC_{1,2,4,5}$  are possible com-

promises for achieving satisfied consumers with a minimal set of features implemented, depending on how thoroughly the cloud provider wants to consider the initial queries. Moreover, we noticed that  $S_1$  clusters with  $S_4$  and  $S_5$  better than  $S_2$ , for example. It should be noted that the more we advance to the right in the graph in Figure 2, the lower the satisfaction level for the initial two queries, but the higher the advantages for other queries, since the FCs are infima of more queries. This is a demonstration example, but when larger amounts of data are analyzed and the complete lattice is considered, the results are naturally even more conclusive.

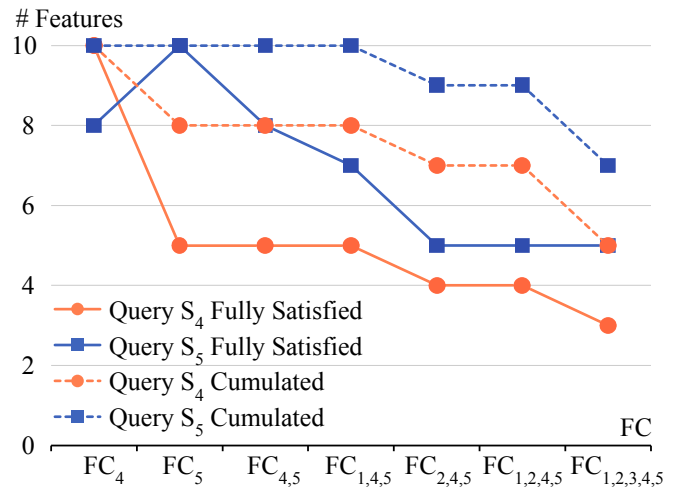


Figure 2. Number of features satisfied for queries  $S_4$  and  $S_5$

#### IV. DISCUSSION

The qualitative results of our approach are promising. Starting the analysis from two advanced search queries for cloud data storage services, our approach was able to identify a potential cluster of queries (e.g.,  $S_1, S_4$  and  $S_5$ ), thus supporting cloud providers in understanding how their potential consumers can be grouped (i). Moreover, new classes of services emerged from the lattice analysis (iii), such as  $FC_{1,4,5}$  and  $FC_{1,2,4,5}$ , showing possible thresholds for achieving satisfied consumers with a minimal set of requirements implemented (ii).

In Section III, we introduced five requirements that should be met by dedicated requirements elicitation methods for the cloud. We now evaluate how our approach satisfies them.

*R1: fit for wider and heterogeneous audiences.* The advanced searches needed by our fuzzy Galois lattices technique are always conducted on cloud providers' or marketplaces' websites. Therefore, our approach allows any number of (potential) consumers from virtually anywhere to input their needs for services. This is done in a completely asynchronous way, without the need of a requirements engineer supervising the requirements elicitation process.

*R2: take less time than traditional elicitation methods.* The approach introduced has a passive character, i.e. consumers are not directly and consciously involved in the requirements elicitation process, since the requirements for new services are inferred based on their searches. This way, virtually no time is dedicated specifically to the elicitation process, but rather to the data analysis.

*R3: make automated elicitation possible.* Our technique is tool-supported, such that most of the analysis is automated. Steps 1-6 are purely automated, and Step 7 is semi-automated. Whereas the new classes of services are automatically generated, while performing the analysis in Step 7, providers can perform a manual what-if analysis to dynamically simulate what happens when only one or a few features are varied, how these impact the general clustering, or zoom in specific parts of the lattice, to analyze the best ideas for new services.

*R4: be applied remotely.* Since this is a search-based approach, it can be applied for any consumers, located anywhere, including those who are not physically reachable. Given its unobtrusive character, the technique is also suitable when consumers would not be able to describe their requirements easily in an interview or a workshop.

*R5: be able to handle volatile requirements.* Our approach enables a continuous elicitation process, since data is collected permanently from consumers who perform advanced searches. This feature is useful for monitoring volatile requirements, which makes the technique especially fitting with the agile character of most cloud provider companies.

There is no perfect, general-purpose requirements elicitation method - each has its strengths and weaknesses and

performs best in a particular context or domain [42]. Our approach is best-suited for the early elicitation phase, and for monitoring market trends. It can be succeeded by more in-depth requirements elicitation with complementary methods such as prototyping and large-scale online experiments. Moreover, having generated the Galois lattice for a set of queries, this can be used by cloud service providers to evaluate where their existing offering is positioned in the spectrum of requested similar services on the market. This can then be used to compute how the existing offering can be enhanced, for example, to fit the needs of a larger population.

A limitation of our approach is that it assumes consumers provide values for all the features specified as advanced search criteria. For example, if a user specifies values of zero or no values for all features, this leads to infima equal to zero, meaning services with no features. This problem can be solved by ignoring the queries having values of zero for all features (outliers) and by allocating default values for all the features with no values assigned.

As far as scalability is concerned, the tool which is currently under development needs three seconds to compute the Galois lattice for 200 advanced search queries for cloud services with ten features, on a 4 GB 1333 MHz DDR3, 1.8 GHz Intel Core i7. A more in-depth analysis of scalability is subject to future work.

#### V. CONCLUSION AND FUTURE WORK

This work presents an approach for inferring new cloud service requirements based on what consumers look for, i.e. their advanced search queries. The approach produces fuzzy Galois lattices, composed of the initial consumer queries and their computed supremum and infimum elements. These lattices can help cloud service providers to analyze how the queries can be grouped to satisfy large populations with a minimum of implemented requirements, and to identify new classes of services needed on the market.

We plan to enhance our approach by releasing a tool that supports the technique introduced, such that cloud providers can benefit from the automatic features. Moreover, we plan to develop a concrete formalism for calculating the satisfaction level for certain combinations of requirements, and to evaluate the approach with real-world data, in order to detect potential shortcomings when working with large datasets from cloud providers.

#### ACKNOWLEDGMENT

This work is supported by the Swiss National Science Foundation (SNSF), grant no. 200021\_150142.

#### REFERENCES

- [1] I. Sommerville and P. Sawyer, *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.
- [2] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and managing software requirements*. Springer, 2005, pp. 19–46.



- [3] N. Maiden and G. Rugg, "Acre: selecting methods for requirements acquisition," *Software Eng Journal*, vol. 11, no. 3, pp. 183–192, 1996.
- [4] T. Tuunanen, "A new perspective on requirements elicitation methods," *JITTA: Journal of Inf Tech Theory & Application*, vol. 5, no. 3, 2003.
- [5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Sys*, vol. 25, no. 6, pp. 599–616, 2009.
- [6] S. Lichtenstein, L. Nguyen, and A. Hunter, "Issues in IT service-oriented requirements engineering," *Australasian Journal of Inf Sys*, vol. 13, no. 1, 2007.
- [7] S. Zardari and R. Bahsoon, "Cloud adoption: a goal-oriented requirements engineering approach," in *2<sup>nd</sup> Intl Workshop on Software Eng for Cloud Computing*. ACM, 2011, pp. 29–35.
- [8] I. Todoran, N. Seyff, and M. Glinz, "How cloud providers elicit consumer requirements: An exploratory study of nineteen companies," in *21st IEEE Intl Requirements Eng Conf (RE'13)*, 2013, pp. 105–114.
- [9] H.-L. Yang and J.-H. Tang, "A three-stage model of requirements elicitation for web-based information systems," *Ind Mgmt & Data Sys*, vol. 103, no. 6, pp. 398–409, 2003.
- [10] W. J. Lloyd, M. B. Rosson, and J. D. Arthur, "Effectiveness of elicitation techniques in distributed requirements engineering," in *IEEE Joint Intl Conf on Requirements Eng*. IEEE, 2002, pp. 311–318.
- [11] R. Akolkar, T. Chefalas, J. Laredo, C.-S. Perng, A. Sailer, F. Schaffa, I. Silva-Lepe, and T. Tao, "Towards cloud services marketplaces," in *8<sup>th</sup> Intl Conf and Workshop on Sys Virtualization Mgmt*, 2012, pp. 179–183.
- [12] R. Belohlavek, "Fuzzy Galois connections," *Mathematical Logic Quarterly*, vol. 45, no. 4, pp. 497–504, 1999.
- [13] D. E. Damian and D. Zowghi, "RE challenges in multi-site software development organisations," *Requirements Eng*, vol. 8, no. 3, pp. 149–160, 2003.
- [14] P. Grünbacher and N. Seyff, "Requirements negotiation," in *Engineering and Managing Software Requirements*. Springer, 2005, pp. 143–162.
- [15] A. Daugulis, "Time aspects in requirements engineering: Or 'every cloud has a silver lining'," *Requirements Eng*, vol. 5, no. 3, pp. 137–143, 2000.
- [16] L. Liu, *Software Reuse in the Emerging Cloud Computing Era: Goal-Based Requirements Elicitation for Service Reuse in Cloud Computing*. IGI Global, 2012.
- [17] D. Villegas and S. M. Sadjadi, "Mapping non-functional requirements to cloud applications," in *SEKE*, 2011, pp. 527–532.
- [18] K. Beckers, S. Fassbender, and H. Schmidt, "An integrated method for pattern-based elicitation of legal requirements applied to a cloud computing example," in *7<sup>th</sup> Intl Conf on Availability, Reliability and Security (ARES)*, 2012, pp. 463–472.
- [19] K. Beckers, M. Heisel, I. Cote, L. Goëke, and S. Guler, "Structured pattern-based security requirements elicitation for clouds," in *8<sup>th</sup> Intl Conf on Availability, Reliability and Security (ARES)*, 2013, pp. 465–474.
- [20] S. L. Lim, D. Quercia, and A. Finkelstein, "Stakenet: using social networks to analyse the stakeholders of large-scale software projects," in *32<sup>nd</sup> ACM/IEEE Intl Conf on Software Eng*, vol. 1. ACM, 2010, pp. 295–304.
- [21] N. Kukreja and B. Boehm, "Process implications of social networking-based requirements negotiation tools," in *Intl Conf on Software and Sys Process (ICSSP)*. IEEE, 2012, pp. 68–72.
- [22] J. D. Herbsleb, "Global software engineering: The future of socio-technical coordination," in *Future of Software Eng*. IEEE Computer Society, 2007, pp. 188–198.
- [23] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-based stakeholder participation in requirements engineering," *IEEE Software*, vol. 24, no. 2, pp. 28–35, 2007.
- [24] C. Solis and N. Ali, "Distributed requirements elicitation using a spatial hypertext wiki," in *5<sup>th</sup> IEEE Intl Conf on Global Software Eng (ICGSE)*, 2010, pp. 237–246.
- [25] P. Liang, P. Avgeriou, and V. Clerc, "Requirements reasoning for distributed requirements analysis using semantic wiki," in *4<sup>th</sup> IEEE Intl Conf on Global Software Eng (ICGSE)*, 2009, pp. 388–393.
- [26] S. Lohmann, T. Riechert, S. Auer, and J. Ziegler, "Collaborative development of knowledge bases in distributed requirements elicitation," in *Software Eng (Workshops)*, vol. 122, 2008, pp. 22–28.
- [27] L. Karlsson, Å. Dahlstedt, J. N. och Dag, B. Regnell, and A. Persson, "Challenges in market-driven requirements engineering - an industrial interview study," in *8<sup>th</sup> Intl Workshop on Requirements Eng: Foundation for Software Quality (REFSQ'02)*, 2003, pp. 101–112.
- [28] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans on Knowledge and Data Eng*, vol. 17, no. 6, pp. 734–749, 2005.
- [29] N. Abdullah, Y. Xu, and S. Geva, "Integrating collaborative filtering and search-based techniques for personalized online product recommendation," in *11<sup>th</sup> IEEE Intl Conf on Data Mining Workshops (ICDMW'11)*, 2011, pp. 711–718.
- [30] T. Q. Lee, Y. Park, and Y.-T. Park, "A time-based approach to effective recommender systems using implicit feedback," *Expert Systems with Applications*, vol. 34, no. 4, pp. 3055–3062, 2008.
- [31] P. Resnick and H. R. Varian, "Recommender systems," *Commun ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [32] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans on Inf Sys (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.
- [33] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. Springer Berlin Heidelberg, 2006, pp. 25–71.
- [34] R. Wille, "Restructuring lattice theory: An approach based on hierarchies of concepts," in *Formal Concept Analysis*, ser. LNCS. Springer, 2009, vol. 5548, pp. 314–339.
- [35] H. Sahraoui, W. Melo, H. Lounis, and F. Dumont, "Applying concept formation methods to object identification in procedural code," in *12<sup>th</sup> IEEE Intl Conf in Automated Software Eng*, 1997, pp. 210–218.
- [36] M. Siff and T. Reps, "Identifying modules via concept analysis," *IEEE Trans on Software Eng*, vol. 25, no. 6, pp. 749–768, 1999.
- [37] B. Fischer, "Specification-based browsing of software component libraries," *Automated Software Eng*, vol. 7, no. 2, pp. 179–200, 2000.
- [38] K. Denecke, M. Ern , and S. L. Wismath, *Galois Connections and Applications*. Kluwer Academic Publishers, 2004.
- [39] M. Ern , J. Koslowski, A. Melton, and G. E. Strecker, "A primer on Galois connections," *Annals of the New York Academy of Sciences*, vol. 704, no. 1, pp. 103–125, 1993.
- [40] B. Ganter, R. Wille, and C. Franzke, *Formal concept analysis: mathematical foundations*. Springer-Verlag New York, 1997.
- [41] I. Todoran, "Stakecloud: Stakeholder requirements communication and resource identification in the cloud," in *20<sup>th</sup> IEEE Intl Requirements Eng Conf (RE'12)*. IEEE, 2012, pp. 353–356.
- [42] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Conf on the Future of Software Eng*. ACM, 2000, pp. 35–46.